



# DECUS

## PROGRAM LIBRARY

DECUS NO.	8-408
TITLE	DISK UTILITY PROGRAM
AUTHOR	P. Galen Lenhert and Douglas Henry
COMPANY	Vanderbilt University Nashville, Tennessee
DATE	January 21, 1971
SOURCE LANGUAGE	PAL-D





## DISK UTILITY PROGRAM

DECUS Program Library Write-up

DECUS NO. 8-408

DUP works with 1 to 4 DF-32 discs and a teletype. The current version will not recognize or use any other devices. It is self-contained and does not need any other programs or subroutines (such as the monitor head.) It performs the functions of dumping and restoring parts of the disc, either by absolute block address or in monitor-type files, erasing files from the disc, and listing the names of all files present.

DUP uses core from 0 to 2177 and has a starting address of 200. (SAVE DUP!0-2177;200).

When entered DUP types OPT-. The legal responses are D,E,F,L,P,R,T, and X. Other responses cause retyping of the request.

X causes a branch to 7600 (monitor restart.)

T(railer) causes DUP to type  $\uparrow$ , wait until  $\uparrow P$  is typed, punch about 12 inches of 200g trailer, then repeat the  $\uparrow$ ,  $\uparrow P$  sequence.

P(unch): After typing the option, type  $\alpha \beta \downarrow$ .  $\alpha$  is the name of a file, or a block address, or 2 block addresses separated by a minus sign.  $\beta$  is either non-existent or is any string of comments, instructions, etc., which begins with a space. After this DUP types  $\uparrow$  and waits for  $\uparrow P$ , dumps the requested area, and again types  $\uparrow$  and waits for  $\uparrow P$ . If  $\alpha$  was two octal numbers, they are the bounds, in block addresses, of the disc area to be dumped. For a description of the tape produced, see below.

E(rase): After typing E, a file name is typed, then  $\downarrow$ . The named file will depart this earth rapidly.

L(ist): The number of free blocks on the disc is typed followed by the names and number of blocks of each file.

R(estore): Turn on the reader and the disc area will be restored. DUP halts after all is back in. Press CONT to get another OPT-. If a checksum error is detected DUP types "C" and halts. Place the first frame of the tape block causing the error over the read station and press continue.

F(iles): Allows the user to enter a list of file names (or block addresses) with comments ( $\alpha_1 \beta_1 \downarrow \alpha_2 \beta_2 \downarrow \dots \alpha_n \beta_n \downarrow \downarrow$ ). The entries are separated by single carriage returns and terminated with 2 carriage returns.

D(ump): Causes all files in the list entered with F to be punched.

All files are handled by name alone without reference to type, therefore, the same file name should not be used for both an ASCII and binary file. If there are 2 identical files, DUP will always pick the first file.

When restoring files DUP will replace the contents of a file that is already present and will not place two consecutive blocks from tape in consecutive disc blocks. This means that it is possible, but not likely, for R(estore) to abort before finishing, as if all space were used, when one block remains free.

The basic tape format is three frames per two words, as described in the Disc Monitor Manual. Each file is preceded by leader and a copy of the dump instruction line and a checksum is punched at the end of each block. If a monitor file is



punched, each disc block is separated from the others by 15 frames of 200 code and the first is preceded by important file information (type, start address, entry point.) "Files" specified by absolute block address have the checksum after each disc block but the blocks are not separated by 200 code.

Both files and monitor blocks (specified by block addresses) are read from and written to the disc as blocks. The link word is lost when files are punched and it is regenerated when they are restored (by looking ahead for free blocks.) When monitor blocks are punched, the "link word" is considered part of the block and is saved and rewritten when the tape is read to restore the disc. Files and monitor blocks can appear on the same tape, in any order, provided that the philosophy of the file directory is not violated by storing monitor blocks into a disc area specified in the SAM blocks as open to files.

#### TAPE FORMAT I (file specified by name)

1. LEADER-TRAILER (200g)
2. FILE NAME terminated by a space (ASCII characters)
3. COMMENT OR IDENTIFICATION terminated by blank CR, LF, CR, LF
4. DIRECTORY INFORMATION (3 paper tape characters for 2 words)
  - a) First character--least significant part of WORD 1
  - b) Second character--least significant part of WORD 2
  - c) Third character--least significant half makes up most significant third of WORD 1, most significant half makes up most significant third of WORD 2.
  - d) WORD 1: The left 6 bits are the starting address for loading (divided by 100g) if the file is contiguous, and 111111 if the file is non-contiguous. The right 6 bits of WORD 1, give the file type as used for word 5 of the DN entry (page B-4 Disc Monitor Manual of July 1969).
  - e) WORD 2 gives the entry point for the file.
5. The first block of the file begins with the next tape character after the 3 directory information characters. The format is 3 tape characters for 2 disc words as described above (a,b,c).
6. Each block ends with a checksum and 15 frames of 200g code.
7. Each block after the first omits the directory information.
8. The last block of the file ends with an ↑ after the checksum.

#### TAPE FORMAT II ("file" saved by absolute disc block address)

Same as above except that the file name is replaced by the address limits in monitor blocks and no directory information is included. Leader-Trailer code is not punched between blocks, only between "files".

Page Zero Variables and Pointers

TT1, TT1 + 1, TT1 + 2, TEMP: Temporary storage

LNBF: Points to the next character in the file name-line buffer

LNNAME: Points to the next character in the file name

LITEMP and LITEM: Temporary indirect address pointers

LBUF: Points to the next word in the I/O buffer (disc to tape or tape to disc) used by the tape read-punch subroutines.

CHEK: Checksum accumulated here

SAMHLF: Zero points to the right half of a SAM word and one points to the left half.

SAMADR: The address of the current SAM word (in SAMBUF). Corresponds to BLOCK after BKTOSM has been called.

FNAME1, FNAME2: File name in DN format

TEMP1, TEMP2, TEMP3: Temporary storage

BLOCK: Block number of monitor block being read, written, searched for, found to be free, etc.

CFLNO: Current file number

NAME1, NAME2: File name in DN format

NFLNO: New file number. Used in loading SAM words.

NFILES: Number of files read in with the F(iles) option

MSMLMT: Minus the number of disc blocks on the system (-number of discs \*400)

TPBLKS: Minus the number of blocks in the "abs file" to be dumped or restored.



### Subroutine Descriptions

#### BKTOSM

BLOCK to SAM conversion sets SAMHLF to point to the right or left half of the SAM word and SAMADR to point to its location in SAMBUF.

USES: BLOCK  
SETS: SAMHLF, SAMADR  
CALLS: R61  
CALLED BY: FNDBLK, RESTOR  
ACCUMULATOR: enter clear, exit clear

#### CKABS

Checks the "file name" pointed to by LNNAME to see if it is an "abs file." If an "abs file" is specified, the first block is stored in BLOCK and minus the number of blocks in TPBLKS. Return is to CALL + 1 for an "abs file" and to CALL + 2 for a monitor file.

USES: LNNAME  
SETS: LNNAME, BLOCK, TPBLKS, TT1, TT1 + 1, TEMP1  
CALLS: R3L  
CALLED by: PFILE, RESTOR  
ACCUMULATOR: exit clear

#### CNTBKS

Counts the number of blocks in the file specified by CFLNO

USES: CFLNO (used by FNDBLK)  
SETS: BLOCK  
CALLS: FNDBLK  
CALLED BY: LIST  
ACCUMULATOR: enter clear, exit clear

#### DNSMWT

Writes the up-to-date core copy of the DN and SAM blocks onto the disc.

USES:  
SETS: TEMP, LITEM, LITEMP  
CALLS: WRITEB  
CALLED BY: RESTOR, ERASE  
ACCUMULATOR: enter clear, exit clear

### DPNCH3

Punches 3 paper tape characters from 2 Buffer words. Picks up the words via LBUF.

USES: LBUF  
SETS: TEMP1  
CALLS: PCHR, R6L  
CALLED BY: PCH200, PFILE  
ACCUMULATOR: enter clear, exit clear

### ECFLNO

Erases an entire file from the SAM. The file to be erased is specified by CFLNO.

USES:  
SETS: BLOCK, NFLNO  
CALLS: FNDBLK, MODSAM  
CALLED BY: RESTOR, ERASE  
ACCUMULATOR: enter clear, exit clear

### FNDBLK

Finds the next block with the file no. CFLNO. Starts the search with the SAM location corresponding to BLOCK. Returns to CALL + 1 if the file is not present and to CALL + 2 if found.

USES: CFLNO, BLOCK  
SETS: BLOCK  
CALLS: BKTOSM, R6R  
CALLED BY: ECFLNO, PFILE, CNTBKS, RESTOR  
ACCUMULATOR: enter clear, exit clear

### GFLNO

Picks up the file name specified by CFLNO and stores it in NAME1, NAME2. Sets and uses LITEMP. If CFLNO points to a free DN entry, exit is to CALL + 1 and LITEMP points to the second part of the file name in DNBUF. If the DN entry is in use, return is to CALL + 2. LITEMP then points to the 3rd word of the DN entry.

USES: CFLNO  
SETS: NAME1, NAME2, LITEMP  
CALLS: \*none-  
CALLED BY: NAMSCH, LIST  
ACCUMULATOR: enter clear, exit clear

### MODSAM

Loads the SAM location specified by SAMADR and SAMHLF with the file number in NFLNO.

USES: SAMHLF, SAMADR, NFLNO



SETS: TEMP1  
CALLS: R6L  
CALLED BY: ECFLNO, RESTOR  
ACCUMULATOR: enter clear, exit clear.

#### NAMSCH

Finds the file number of the file specified in FNAME1 and FNAME2 or the next free file. Returns to CALL + 1 for no match and CALL + 2 for a match. CFLNO is the number of the matching file or the blank file if no match is found. CFLNO is set to zero if no match is found and all DN entries are full. TEMP3 points to the empty DN entry if one is found.

USES: FNAME1, FNAME2  
SETS: CFLNO, TEMP1, TEMP2, TEMP3  
CALLS: GFLNM  
CALLED BY: PFILE, RESTOR, ERASE  
ACCUMULATOR: exit clear

#### NMTOFN (and PKCKSP)

Converts typed name pointed to by LNNAME to a monitor file name in FNAME1 and FNAME2.

USES: LNNAME  
SETS: TEMP2, LNNAME, TEMP1, FNAME1, FNAME2  
CALLS: R6L  
CALLED BY: PFILE, RESTOR, ERASE  
ACCUMULATOR: enter clear, exit clear

#### OCTPRT

Prints, as an octal number, the contents of the accumulator. CALL + 1 contains minus the no. of digits requested. Returns to CALL + 2.

USES:  
SETS: TEMP1, TEMP2, TEMP3  
CALLS: R3L, PCHR  
CALLED BY: LIST  
ACCUMULATOR: enter with no. to print, exit clear

#### PCH200

Punches 2N words from the disc as 3N paper tape characters plus a checksum. Enter with -N in accumulator

USES:  
SETS: CHECK, TEMP2, IOBUF, LBUF  
CALLS: DPNCH3, PCHR  
CALLED BY: PFILE  
ACCUMULATOR: enter with triplet count (-N), exit clear



### PFILE

Punches a single file including leader, file name and identification. Everything up to but not including the "file mark" (↑).

USES: LNBF, LITEMP

SETS: LNNAME, LNBF, LITEMP, IOBUF, LBUF, BLOCK, TPBLKS

CALLS: LTPNCH, PLINE, NMTOFN, NAMSCH, R6R, DPNCH3, FNDBLK, READB, PCH200, CRLF, PCHR, CKABS

CALLED BY: PUNCH, DUMP

ACCUMULATOR: enter clear, exit clear

### PLINE

Picks up a file line (name and identifying information) via LNBF and prints it.

USES: LNBF

SETS:

CALLS: PCHR, CRLF

CALLED BY: PFILE

ACCUMULATOR: enter clear, exit clear

### PNAME (and CHROUT)

Prints a file name stored in NAME1, NAME2

USES: NAME1, NAME2

SETS: TEMP1, TEMP2

CALLS: R6R, PCHR

CALLED BY: LIST

ACCUMULATOR: exit clear

### RD200

Fills IOBUF from paper tape. 2N words from 3N paper tape characters. Reads checksum from tape and compares it with calculated checksum. Returns to ERR1 if checksums don't match. Enter with -N in the accumulator.

USES:

SETS: LBUF, CHEK

CALLS: TREAD3

CALLED BY: RESTOR

ACCUMULATOR: enter with triplet count (-N), exit clear

### RDDNSM

Reads the DN and SAM blocks into DNBUF and SAMBUF. An up-to-date copy is kept in core from the time the program is entered.

USES:

SETS: BLOCK

CALLS: R6L, READB

CALLED BY: AT ENTRY ONLY

ACCUMULATOR:               exit clear

RDLINE

Reads a file name and identifying information from paper tape. Rejects all characters preceding the file name which are greater than 333 and less than 240. Exits to WAIT if more than 48 leader characters are found. File name and line are stored into the line buffer via LNBF. Each line is terminated by a carriage return which causes a nul character to be entered into the buffer and a return to CALL + 1. If a file name is started with a carriage return, the subroutine returns immediately to OPT-.

USES:           LNBF, NFILES

SETS:           NFILES, LNBF

CALLS:          RCHR

CALLED BY: FILES, RESTOR, ERASE

ACCUMULATOR:       exit clear

READB

Reads a monitor block from the disc to core. The block to be read is specified by BLOCK, the core address by CALL + 1. Return is to CALL + 2 with the link word in the accumulator. ERROR stop at DSKOP + 4

USES:           BLOCK

SETS:

CALLS:          R6R

CALLED BY: PFILE, RDDNSM

ACCUMULATOR:   enter clear, exit with link word

TREAD3

Reads 3 paper tape characters and converts to 2 words stored via LBUF

USES:           LBUF

SETS:           TT1, TT1 + 1, TT1 + 2

CALLS:          RCHR

CALLED BY: RD200, RESTOR

ACCUMULATOR:   exit clear

WRITEB

Writes one page of core to the disc. The block address on the disc is specified by BLOCK, the core address by CALL + 1. The link word is in the accumulator at entry. ERROR stop at DSKOP + 4.

USES:           BLOCK

SETS:

CALLS:          R6R

CALLED BY: RESTOR

ACCUMULATOR:   enter with link word